Electronic Design

By ELISA BERTINO, CS Department, Purdue University,
https://www.cs.purdue.edu/people/faculty/bertino.html

Identifying Vulnerabilities in Cellular Networks

This article takes an in-depth look at a systematic framework for the analysis of cellular-network protocols, involving a 4G LTE example, to enhance security.

Cellular networks are a critical infrastructure supporting applications in all domains we may think of, ranging from e-commerce, transportation/mobility, and education to eHealth/personal well-being, and manufacturing. Cellular networks will be a key infrastructure for Internet of Things devices, and, indeed, the vision is that next-generation cellular networks will be more about devices than people. In other words, the goal of next-gen cellular networks is to be "sensing networks."

However, even without looking into what next-gen cellular networks will be, it's clear that technologies for cellular networks have made major advances in the past few years. Fourth Generation Long-Term Evolution (4G LTE) technology has increased the bandwidth available for smartphones, in essence delivering broadband capacity.

The most recent 5G technology further enhances transmission capacity and reduces latency, energy consumption, and error rates through the use of several technologies, including millimeter waves; small cells; massive multiple-input, multiple-output antennas; beamforming; full duplex transmission; and software-defined networks (SDNs). It enhances the flexibility of cellular networks by separating network control and forwarding planes and making the control plane directly programmable.[1]

Because cellular networks are pervasive and used in sensitive applications, their security is a critical requirement. For example, a denial-of-service attack against a cellular network may paralyze communities and service infrastructures, with disastrous consequences.

Securing cellular networks is a challenging task because of their complexity. Cellular networks consist of multiple layers—e.g., physical layer, radio-resource-control (RRC) layer, non-access stratum (NAS) layer, etc. Each layer, in turn, has its own protocols to implement its procedures, such as the protocols for attaching/detaching devices to/from the network and for paging devices notifying of incoming voice calls and Short Message Service (SMS) text messages. Additional requirements, such as backward compatibility and interoperation across different wireless communication technologies, add to the complexity.[1]

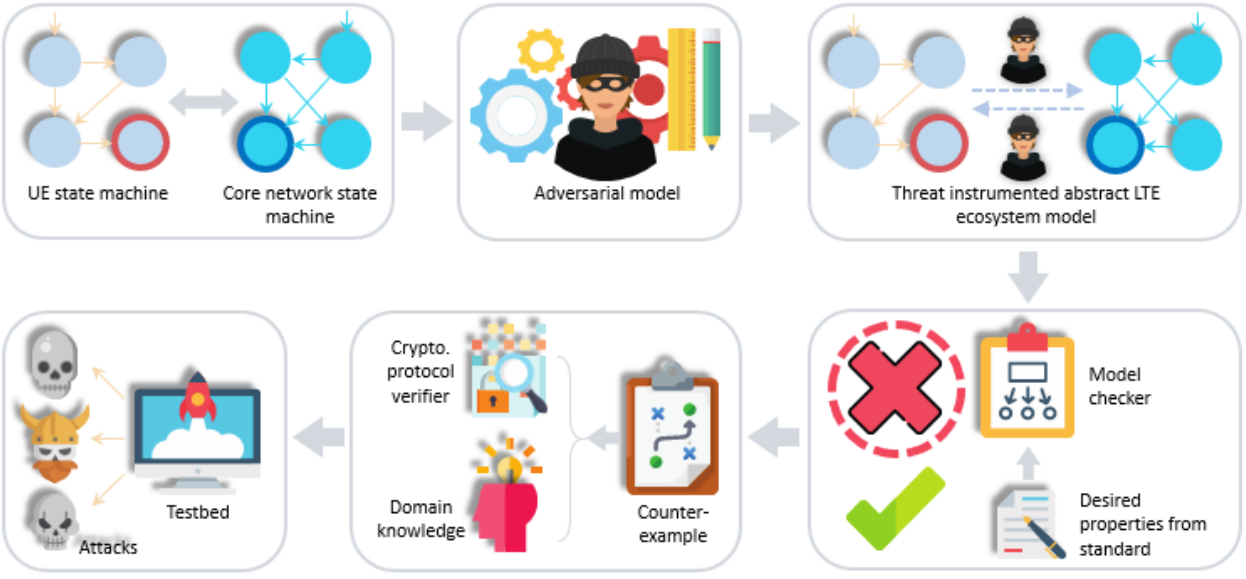
Comprehensive approaches to protecting cellular networks require deploying a wide variety of security techniques, ranging from basic techniques such as encryption and digital signatures, to software patching, anomaly detection, network segmentation, device hardening, etc. *(see Reference 2 for an example of security measures for network infrastructure devices).* However, a critical prerequisite to securing cellular networks is that the protocols designed, implemented, and deployed in them must be free of vulnerabilities. Due to the complexity of those protocols, systematic methodologies for their analysis are required.

Methodologies for Verifying Cellular Network Protocols

Perhaps the first systematic methodology for analyzing cellular network protocols is the LTEInspector methodology,[2] developed for the analysis of the NAS layer of the 4G LTE protocol stack. This layer manages the establishment of communication sessions and maintains continuous communications with the user equipment (UE), i.e., the cellular phone, as it moves.

The NAS layer provides a set of protocols governing the interactions between the UE and the core nodes, such as the mobile switching center, serving GPRS support node, or mobility management entity (MME). We refer to the set of core nodes as "core network" (CN). Each such protocol consists of multiple steps. For example, the protocol for UE attach includes the following high-level steps:

1. The UE sends an attach request to the CN, providing its security capabilities.
2. A mutual authentication is executed between the UE and the CN.
3. If authentication is successful, the UE and the CN



1. As depicted in this architecture, the input to the LTEInspector methodology is a representation of each protocol in terms of two finite state machines, the user equipment, and the core network. (Image from Reference 3)

negotiate the algorithms to use for encryption and digital authentication.

4. Once the negotiation is completed, the CN sends the UE an accept attach message.

5. The UE confirms the attach.

Attacks and other failures may happen during any such step. The goal of the LTEInspector methodology is to analyze multi-step protocols to identify vulnerabilities in these steps.

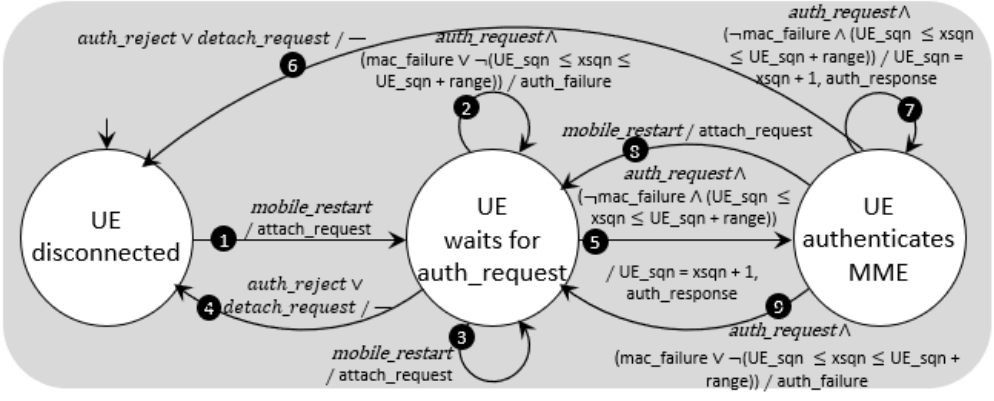
The input to the LTEInspector methodology is a representation of each protocol in terms of two finite state machines (FSMs), one for each party involved in the protocol—that is, the UE and the CN *(Fig. 1)*. An FSM is an abstract machine that can be in exactly one among a finite number of states at any given time. The FSM can change from one state to another in response to some input; these changes are referred to as "transitions." Preconditions also can be associated to transitions; in such cases, for the input to trigger a transition, the preconditions must be true. Also, as part of a transition, actions can be executed.

FSMs are, thus, a representation well-suited for the analysis of multistep protocols. An example FSM modeling the attach protocol at the UE side is shown in *Figure 2*. In the example, we use the MME as party, from the CN, involved in this protocol.

From the diagram, we can see that the UE is initially in a disconnected state and then, upon the phone restart (indicated by the condition *mobile_restart*), the UE sends an attach request to the MME and transitions to the state in which it waits for the authentication request (that is, the state *UE waits for auth_request*). Once the UE is in this state, different transitions can happen. For example, the UE is restarted (Transition #3) or the authentication fails (Transition #4), or the authentication of the UE by the MME is successful and, as a result, the UE moves to state in which it authenticates the MME (Transition #5).

Thus, the goal of the LTE methodology is to determine scenarios (i.e., sequences of transitions) in which the UE, because of attacks, is unable to reach the final intended state (i.e., the state in which the UE has been authenticated by the MME and the MME has been authenticated by the UE).

To identify such attacks, one must consider the capabili-



2. A finite state machine—such as this simplified example UE FSM modeling the attach protocol—is a representation well-suited for the analysis of multistep protocols. (Image from Reference 3)

ties of the attacker, referred to as "adversarial model," that are relevant to the protocols to be analyzed. Because the focus of LTEInspector is on vulnerabilities in communication protocols (and not, for example, on vulnerabilities in the equipment hardware), LTEInspector adopts the Dolev-Yao attack model.[4]

Under the Dolev-Yao attack model, the capabilities of the attacker include dropping/modifying messages exchanged on the network, injecting false messages, impersonating legitimate parties in communications, and eavesdropping messages. Also, under this attack model, the attacker adheres to the assumption that the attacker is unable to decrypt messages without possessing the proper decryption keys and cannot forge the digital signatures of legitimate parties without possessing the keys used for the signature.

The FSM extended with the inclusion of the adversarial model is then given as input to the NuSMV model checker,[5] together with properties to be verified. Then, by using an iterative process known as "property refinement," each property of interest is verified separately.

In property refinement, additional conditions are added to the property to be verified to exclude spurious cases that can lead to property violations, but which do not represent vulnerabilities. If no violations are detected, the property is considered as verified by the protocol.

On the other hand, when there's a violation, the model checker returns the scenario that leads to the violation of the property. The scenario is then analyzed via a cryptographic verifier to determine whether the violation still occurs under the cryptographic assumptions about the attacker. If this is the case, as a last step, the attack is checked in an actual testbed with commercial UEs to determine whether the attack is possible in practice, as commercial UEs may implement additional defenses.

By using LTEInspector, several new vulnerabilities were identified that can be exploited in actual attacks *(see Reference 3 for details on these attacks)*. Most such vulnerabilities were due to the lack of deployment of well-known security techniques for some messages exchanged by the protocol. Examples include lack of replay protection and lack of digital signature for certain broadcast messages, such as the paging messages broadcast by cellular towers to notify UEs of calls and SMS.[6]

Starting from LTEInspector, other notable methodologies were designed, including 5GReasoner,[7] which extends LTEInspector by modeling, in addition to the NAS layer, RRC, and a fuzzing-based approach to identifying design and implementation vulnerabilities in 5G code by carriers and device vendors.[8]

Creating FSM Models of Communication Protocols

The application of formal methodologies, like LTEInspector and 5GReasoner, requires formal models of the protocols to be analyzed. Based on our experience, such a model can be extracted from the standardization documents, as in the case of the model developed for LTEInspector, or from the protocol implementations. Both approaches have challenges.

Extracting models from standardization documents requires a huge amount of manual effort, as these documents are often very large and convoluted. Addressing this issue would require the design of specialized natural language processing approaches, perhaps based on artificial-intelligence/machine-learning techniques.

One advantage, though, is that the analysis performed on such models allows one to identify errors and ambiguities in the standardization documents or other specification documentation. Indeed, among the new vulnerabilities found by LTEInspector, more than half were due to issues in the specification from standardization documents.

Extracting models from actual implementations of the protocols has the advantage that automatic or semi-automatic approaches can be used, such as the recent ProChecker methodology.[9] ProChecker leverages the testing infrastructure used for the code to extract from the implementation an FSM model of the protocol. Because the model is extracted from the implementation, it's more fine-grained than a model obtained from the natural language specification.

Using a more detailed model allows one to identify vulnerabilities that aren't identified by more abstract models. For example, when applying ProChecker to an industrial codebase that has a size of around 80 GB, implementation of the NAS layer identified three new protocol attacks. These weren't identified in analysis by LTEInspector using a more abstract model extracted from the standardization documents, as well as six implementation issues.

One disadvantage is that the extracted model may be very large, which results in scalability issues with the formal analysis tools used. Furthermore, the model may contain unnecessary details that make it more difficult for the programmers/software engineers to understand the vulnerabilities.

Key Insights

Ensuring that cellular network protocols are free of vulnerabilities of varying nature is a challenging task that requires the use of several techniques. For example, memory vulnerabilities, such as buffer overflows, are today well understood and identified in various ways, such as by fuzzing testing. On the other hand, logical vulnerabilities, e.g., lack of digital signatures on messages, are more difficult to identify. Formal verification methodologies, like the ones discussed earlier, are more suitable for identifying these vulnerabilities.

However, the use of these methodologies requires exten-

sive domain knowledge to determine the proper abstraction level(s) for the model and the properties. In general, having an abstract model is useful in defining an initial set of relevant properties and verify whether the protocol, as initially specified, has vulnerabilities. Then a more detailed model can be extracted from the implementation and analyzed by refining the properties defined for the abstract model. In addition, the model extracted from the implementation can be compared with the abstract model to detect whether the implementation is noncompliant with the specification.

To conclude, we have promising techniques and methodologies, and, hopefully, research by industry and academia will enhance and engineer them for practical use.

Elisa Bertino is Samuel D. Conte Professor of Computer Science at Purdue University. Prior to joining Purdue in 2004, she was a professor and department head at the Department of Computer Science and Communication of the University of Milan. She has been a visiting researcher at the IBM Research Laboratory (now Almaden) in San Jose and at Rutgers University, and a Visiting Professor at the Singapore National University and the Singapore Management University.

Elisa is a Fellow member of IEEE, ACM, and AAAS. She received the 2002 IEEE Computer Society Technical Achievement Award for "outstanding contributions to database systems and database security and advanced data management systems," the 2005 IEEE Computer Society Tsutomu Kanai Award for "pioneering and innovative research contributions to secure distributed systems," the 2019–2020 ACM Athena Lecturer Award, and the 2021 IEEE Innovation in Societal Infrastructure Award.

She led the design of the LTEInspector framework to test security properties of cellular networks, leading to the identification of 10 novel vulnerabilities in the 4G LTE standard, as well as the discovery of new privacy attacks in 4G and 5G cellular protocols. For this work, she was named to the GSMA Mobile Security Research Hall of Fame.

References

1. E. Bertino, S. R. Hussain, O. Chowdhury, "5G Security and Privacy: A Research Roadmap." CoRR abs/2003.13604 (2020).

2. Cybersecurity & Infrastructure Security Agency (CISA), "Securing Network Infrastructure Devices." Last accessed on 7/27/2021 at https://us-cert.cisa.gov/ncas/tips/ST18-001.

3. S. R. Hussain, O. Chowdhury, S. Mehnaz, E. Bertino. "LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE." Proceedings of NDSS 2018.

4. D. Dolev and A. C. Yao, "On the security of public key protocols." Stanford, CA, USA, Tech. Rep., 1981, http://www.ncstrl.org:8900/ncstrl/servlet/search?formname=detail&id=oai

5. https://nusmv.fbk.eu/

6. A. Singla, S. R. Hussain, O. Chowdhury, E. Bertino, N. Li, "Protecting the 4G and 5G Cellular Paging Protocols against Security and Privacy Attacks." Proc. Priv. Enhancing Technol. 2020(1): 126-142.

7. S. R. Hussain, M. Echeverria, I. Karim, O. Chowdhury, E. Bertino, "5GReasoner: A Property-Directed Security and Privacy Analysis Framework for 5G Cellular Network Protocol." Proceedings of ACM CCS 2019: 669-684.

8. H. Kim, J. Lee, E. Lee, and Y. Kim, "Touching the Untouchables: Dynamic Security Analysis of the LTE Control Plane." Proceedings of the IEEE Symposium on Security & Privacy (S&P), 2019.

9. I. Karim, S. R. Hussain, E. Bertino, "ProChecker: An Automated Security and Privacy Analysis Framework for 4G LTE Protocol Implementations." Proceedings of the 2021 IEEE ICDCS.